Northeastern University - Seattle

Khoury College of Computer Sciences

CS6200 Information Retrieval Fall 2019

Lecture 2: Search Engine Architecture

Sep 9, 2019

Overview

SEARCH ENGINE ARCHITECTURE

USER'S VIEW OF A SEARCH ENGINE

SEARCH ENGINE: BEHIND THE SCENES

OVERVIEW OF SEARCH ENGINE COMPONENTS

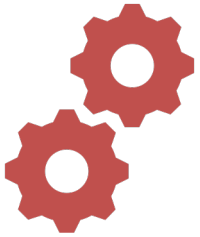
Search Engine “Architecture”

An abstracted view

Various software components

Their interfaces

Relationships between them

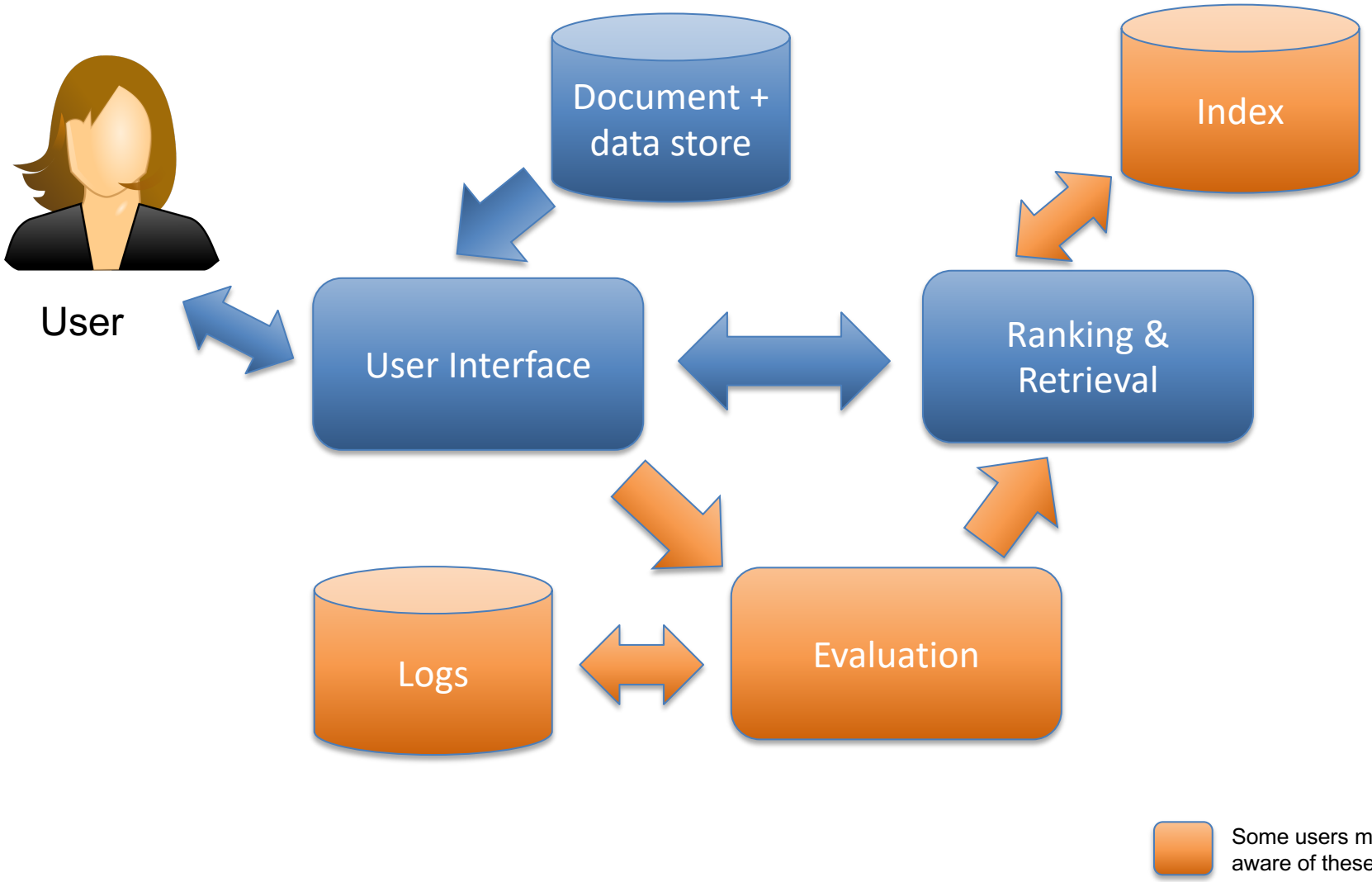


Two key requirements:

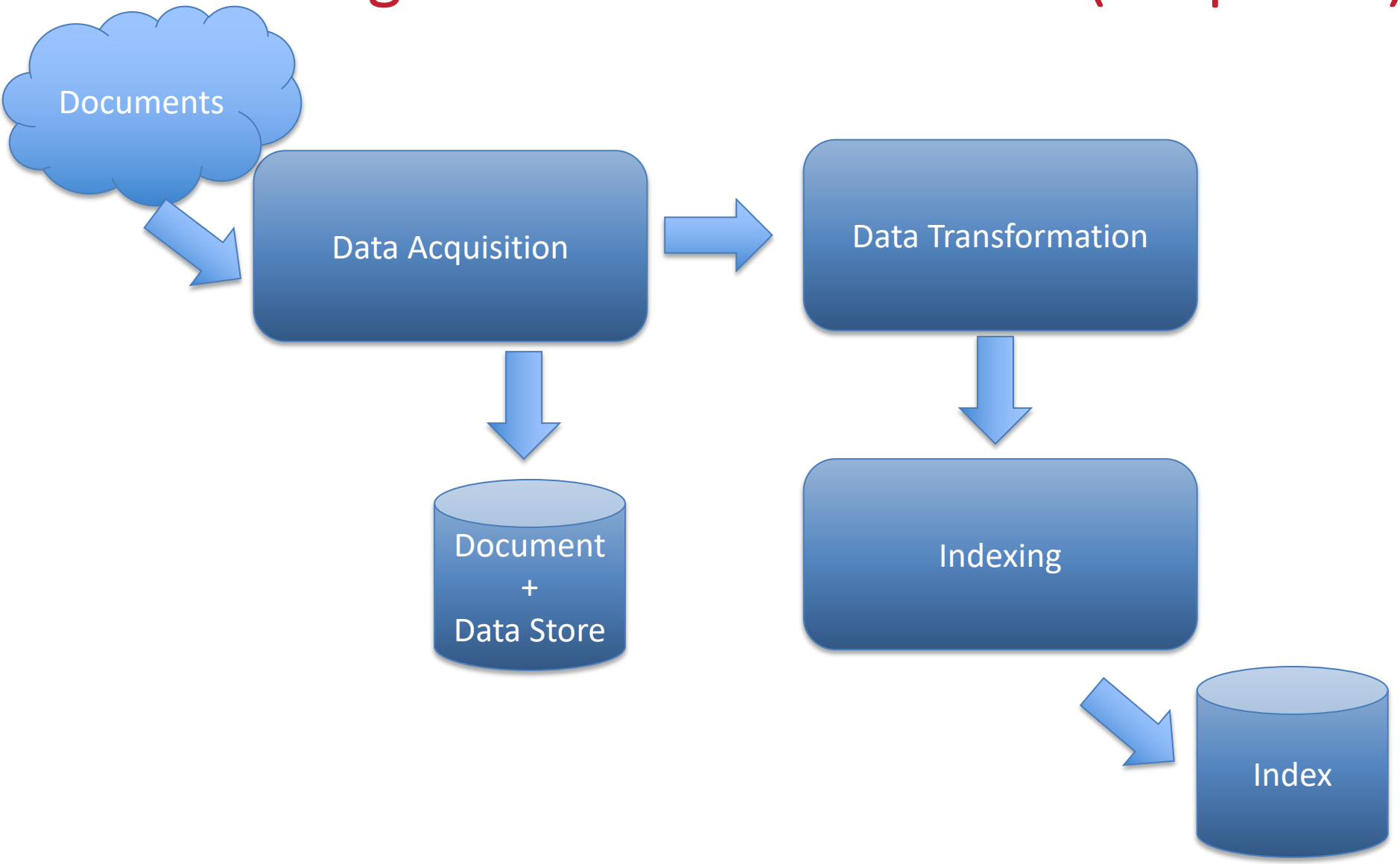
Effectiveness: Quality of the results obtained from the engine

Efficiency: Related to response time, query throughput etc.

Search Engine: User's View (simplified)



Search Engine: Behind the Scenes (simplified)



Search Engine: Key Steps

WEB CRAWLING

NORMALIZING & TRANSFORMING DATA

INDEXING

RANKING & RETRIEVAL

USER INTERACTION

EVALUATION

Coming up: a simplified overview of each component
More details in later lectures!

WEB CRAWLING

Web Crawling: Acquiring Data ..1

- Data can come from *crawlers* looking for data, or *feeds* supplying data to the engine
- Crawlers:
  - Desktop or intranet crawlers: look inside a computer or machines within a company; may look at file folders, databases, SharePoint etc.

Web Crawling: Acquiring Data ..2

- Crawlers:
  - Web crawlers:
    - follow web links across the Web to get new documents,
      - ensure enough documents crawled to get good 'coverage',
      - and these are crawled again with sufficient frequency to keep the index 'fresh'
  - Focused (topic-specific) crawlers
    - Look only for documents of certain content or types (e.g. academic search, images)

Web Crawling: Acquiring Data ..3

- Feeds
  - Collections or streams of documents ‘fed’ to search engine
  - Often done in collaboration with a search engine to support a feature, say product search
  - Real-time feeds useful for news, blogs etc.

Web Crawling Acquiring Data ..4

- Feeds
  - Good for both data owner and search engine:
    - Data owner gets data to the search engine as and when it is updated, and sends out all the data it wants to be seen (for freshness and coverage)
    - Search engine does not have to crawl repeatedly; avoids missing data or crawling data long after update

DATA NORMALIZATION & DATA TRANSFORMATION

Data normalization ..1

- Data acquired from crawls or feeds normalized:
  - Text converted into standard form with meta-data (e.g. HTML, Word docs, PDFs all go to XML or JSON)
  - Unicode standardization
  - Images re-sized, audio/video transcoded

Data normalization ..2

- Data normalized:
  - Metadata such as size, creation date, crawled date, feed info etc. stored
  - Links, anchor text, entities etc. extracted and stored
  - Duplicates/spam etc. identified and rejected

- Data + metadata stored for fast access and processing by other components

Data transformation ..3

[For text:]

- Dealing with ‘stop words’ or ‘function’ words:
  e.g. a, an, the, and, or, from, into, to, be …
  [in English]
  - Huge part of all text input (>~ 40%)
  - Not always good to remove them,
    e.g. “To be or not to be”

Data transformation ..4

[For text:]

- Dealing with punctuation
  - AT&T , .NET, C# , #hashtags, @twitterhandles, …
- Stemming/morphological analysis

  Getting root word from word variants
  - E.g. smile, smiled, smiles, smiling …
  - Not always easy or helpful, depends on language

Data transformation ..5

Internet search engines themselves predate the debut of the Web in December 1990. The Who is user search dates back to 1982 [1] and the Knowbot Information Service multi-network user search was first implemented in 1989.[2] The first well documented search engine that searched content files, namely FTP files was Archie, which debuted on 10 September 1990.[*citation needed*]

From https://en.wikipedia.org/wiki/Web_search_engine

- Anchor text (e.g. “Who is”, “FTP” above) extracted and used as metadata for the site pointed to
- Links extracted to form web graph and in ranking algorithms such as PageRank (popularity/hub info)

Data transformation ..6

- Entities (people, places, company names, SSNs, email addresses, phone numbers, tweet handles etc.) extracted for specific applications .

The parrots of Southeastern Peru crave an earthy delicacy: dirt. At the Colorado clay lick, a cliff face rising above the Tambopata River in the western Amazon Basin, parrots — often hundreds at a time from up to 18 species — gather each day to feast on sun-hardened clay.

"It's a real spectacle of both sight and sound," says biologist Donald Brightsmith of Texas A&M University.

From http://www.npr.org/sections/thesalt/2017/09/07/547981850/why-do-parrots-and-people-eat-clay

INDEXING

Indexing Data ..1

- Main portion: Processes terms in documents to create term-document relationships
  - e.g. Document 1: “The cat sat on the mat”
    Document 2: “The rat sat on the cat”
  - Simple version →

Term	Document 1
The	1
cat	1
sat	1
on	1
mat	1
rat	0

- Since data is fast-changing, must handle updates well
- Sparse matrices? Compression useful.

Indexing Data ..2

- Data collected for ranking, special queries (e.g. proximity*) too:
  - Term statistics
  - Word positions
  - ‘idf’ values (more about this soon!)

- Think:
  What would we index for images/audio/video?

* Proximity search lets you search for two or more terms within a specified distance, e.g. within 4 words of each other

Indexing Data ..3

- Distributed indexes used for huge collections, e.g. Bing/Google
  - Index size and performance limitations
  - Need multiple computers, and/or multiple sites, …
  - Documents split by language or just randomly

RANKING & RETRIEVAL

Ranked Retrieval

- Ranking algorithm: Computes how closely documents match query, and ranks results accordingly
- Many retrieval models, ranking algorithms
- Ranking could be distributed
  e.g. if based on distributed indexes
  - Farm out queries and collate results

- More soon!

USER INTERACTION

User Interaction ..1

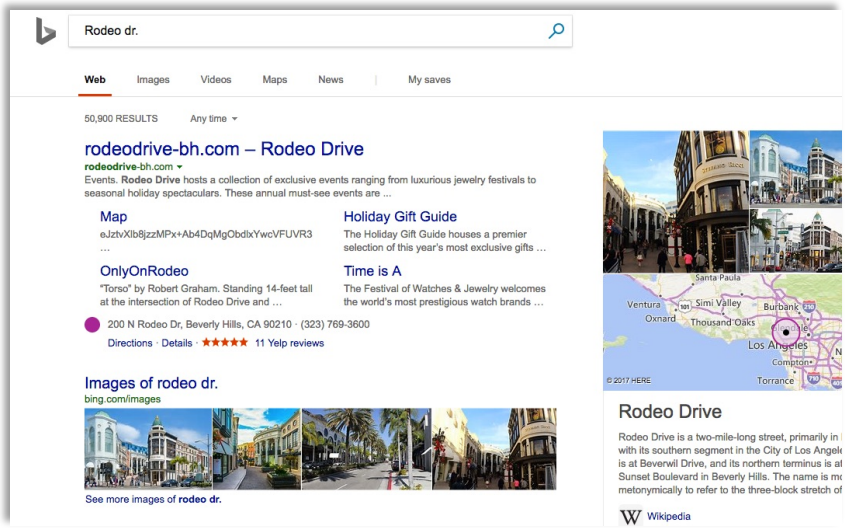
- Input queries
- Transform queries
- Output results

User Interaction ..2

- Input queries may be:
  - Simple free text
  - Boolean queries: [“Roger Federer” AND Wimbledon], [“Bianca Andreescu” AND “US Open Women’s singles”]
  - Include query operators, such as phrase quotes, + to indicate required term, - to indicate term should not occur etc.
- Query parsing required

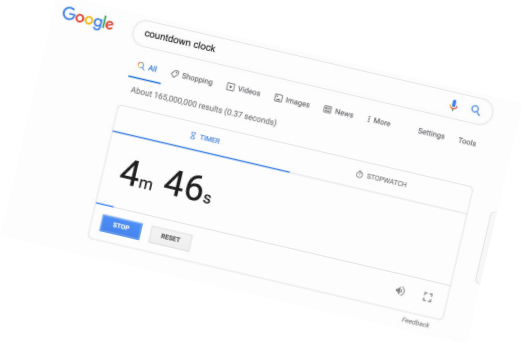
User Interaction ..3

- Queries may be transformed:
  - Spelling correction
  - Query suggestion
  - Expanded with new terms
  - Clarified ([Rodeo Dr.] -> [Rodeo Drive])
    - But when is Dr = Doctor ?



From https://www.bing.com/search?q=Rodeo+dr.

User Interaction ..4



- Results displayed for each query:
  - In ranked order
  - With snippets/hit-highlights to show context of query match
  - With spell suggestions, query suggestions, related documents, grouped/clustered results, advertisements…
  - Right-pane info
  - Instant answers

From https://www.google.com/search?q=Gerry+Salton, https://www.google.com/search?q=100+CAD+in+USD

EVALUATION

Search Engine Evaluation

- Metrics computed to ensure that any new features or code changes only improves the engine
  - Better effectiveness
  - More performant system
- Queries, results and clicks logged
  - To get data for query and spelling suggestions, ranking, advertising relevance, spam detection…

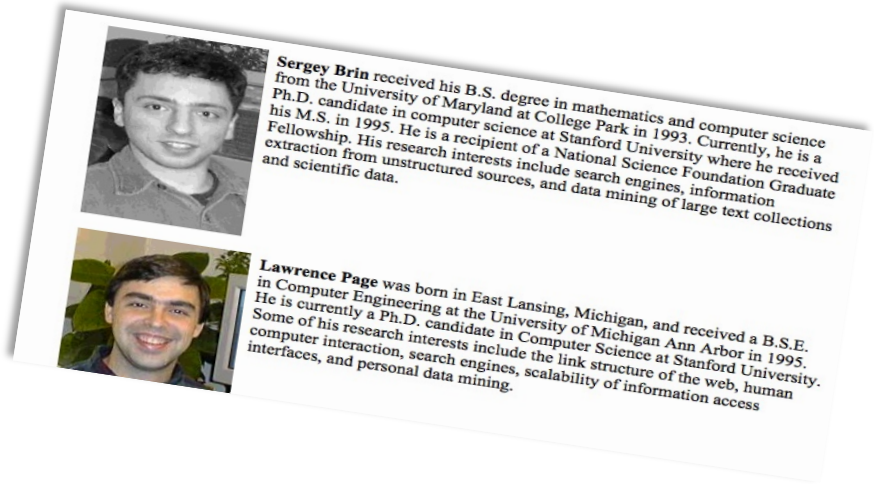
Caveats

- This overview omitted a lot of detail, simplified to get a high-level picture
  - E.g. language identification, duplicate detection, sentiment analysis in product search etc.
- More details in the following lectures
  - Covering main approaches
  - With pointers to other approaches

Summary

- Got a closer look at Search Engines, with an overview of main components
- Next: Acquiring Data

Readings



Sergey Brin received his B.S. degree in mathematics and computer science from the University of Maryland at College Park in 1993. Currently, he is a Ph.D. candidate in computer science at Stanford University where he received his M.S. in 1995. He is a recipient of a National Science Foundation Graduate Fellowship. His research interests include search engines, information extraction from unstructured sources, and data mining of large text collections and scientific data.

Lawrence Page was born in East Lansing, Michigan, and received a B.S.E. in Computer Engineering at the University of Michigan Ann Arbor in 1995. He is currently a Ph.D. candidate in Computer Science at Stanford University. Some of his research interests include the link structure of the web, human computer interaction, search engines, scalability of information access interfaces, and personal data mining.

1. Chapter 2 CMS

2. “The” Google paper:

The Anatomy of a Large-Scale Hypertextual Web Search Engine, Sergey Brin and Lawrence Page, Computer Networks and ISDN Systems 30 (1998) pp. 107- 117.
http://snap.stanford.edu/class/cs224w-readings/Brin98Anatomy.pdf

An earlier version is available at: http://infolab.stanford.edu/pub/papers/google.pdf

Questions?